Name (print): ________________________________

- **INSTRUCTIONS:**
  - Keep your eyes on your own paper and do your best to prevent anyone else from seeing your work.
  - Do NOT communicate with anyone other than the professor/proctor for ANY reason in ANY language in ANY manner.
  - This exam is closed notes, closed books, and no calculator.
  - Turn all mobile devices off and put them away now. You cannot have them on your desk.
  - Write neatly and clearly indicate your answers. What I cannot read, I will assume to be incorrect.
  - Stop writing when told to do so at the end of the exam. I will take 5 points off your exam if I have to tell you multiple times.
  - Academic misconduct will not be tolerated. Suspected academic misconduct will be immediately referred to the Emory Honor Council. Penalties for misconduct will be a zero on this exam, an F grade in the course, and/or other disciplinary action that may be applied by the Emory Honor Council.

- **TIME:** This exam has 8 questions on 12 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

---

*I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Emory community. I have also read and understand the requirements and policies outlined above.*

Signature: ________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>9</td>
<td>24</td>
<td>10</td>
<td>14</td>
<td>9</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (9 points) Variable types and kinds. Consider the following code:

```java
public class Code1 {
    static int p = -2;
    public static void oddManipulation(int a, int b, int c) {
        int temp = a;
        b = temp;
        a = b;
        System.out.println("a: " + a);
        for (int i = 0; i < 1; i++) {
            System.out.print(a + 2 + " ");
        }
        p = a + q;
    }

    static int q = 1;
    public static void main(String[] args) {
        int x = 10, y = 11;
        int z = 9 - x;
        oddManipulation(x, y, z);
        System.out.println("x: " + x + " y: " + y + " z: " + z);
        System.out.println("p: " + p + " q: " + q);
    }
}
```

1. (1 pt) List all local variables in the `oddManipulation` method.

2. (1 pt) List all parameter variables in the `oddManipulation` method.

3. (1 pt) List all class variables in the program above.

4. (3 pts) What would be the output of the program if parameters were passed by `value`?

5. (3 pts) What would be the output of the program if parameters were passed by `reference`?
2. (24 points) For each code fragment below, provide the output of the code. If the code contains errors, find ALL ERRORS in that code fragment. Assume that the method Arrays.toString(arrayName) will return a String containing the elements of the parameter array delimited/separated by commas. The Java library method Arrays.toString(params) are overloaded so it can accept a parameter array of any type (char[], int[], double[], boolean[] and etc).

E.g.

```
int[] x = {12, 23, 54, 11};
System.out.println(Arrays.toString(x));
```

will output 12, 23, 54, 11 to the terminal.

a) Code:

```java
public class Code2A {
    public static void main(String[] args) {
        double[] c = {12.0, 1.1, 3.1, 2.1};
        System.out.println(Arrays.toString(c));

        double[] d = {3.0, 2.0, 1.0};
        for (int k = d.length - 1; k > 0; k--) {
            d[k] = c[k];
        }
        System.out.println(Arrays.toString(d));
    }
}
```

Output:

b) Code:

```java
int[] arr = {1, 2, 0, 1, 9};
for (int i = 1; i < arr.length; i++) {
    if (arr[i] <= arr[i + 1]) {
        System.out.println(i);
    }
}
System.out.println(Arrays.toString(arr));
```

Output:
c) Code:

```java
public class Code2C {
    public static void main(String[] args) {
        int[] a = {1, 4, 7};
        int[] b = new int[3];
        int[] c = new int[1];
        c = a;
        a = b;
        switch (a[0]) {
            case 0: System.out.println(Arrays.toString(b));
            case 1: System.out.println(Arrays.toString(c));
            default: System.out.println(Arrays.toString(a));
        }
    }
}
```

Output:

```
```

d) Code:

```java
public class Code2D {
    public static void main(String[] args) {
        int[] arr = {1, 2, 0, 1, 9};
        for (int i = 1; i < arr.length; i++) {
            arr[i] = arr[i] * arr[i + 1];
        }
        System.out.println(Arrays.toString(arr));
    }
}
```

Output:

```
4
```
3. (10 points) Algorithms:

Assume, as in question 2, that the method Arrays.toString(arrayName) will return a
String containing the elements of the parameter array delimited/separated by commas.
Consider the following 2 arrays:

int[] a = {1000001, 145, 130, 131, 129, 121, 21, -20, -40};
int[] b = {-10, 11, 21, 65, 71, 70, 112, 123};

a) (2 pts) For each array a and b, can the linear (sequential) search algorithm be
performed on the array? If yes, list the elements in the order we will inspect when
searching for value 21 in that array. If not, explain why.

b) (2 pts) For each array a and b, can the binary search algorithm be performed on
the array? If yes, list the elements in order we will inspect when searching for value
22 in that array. If not, explain why.
c) (6 pts) Sorting. What is the output of the following program?

Notice that there are S.o.p statements in the selection sort method.

```java
public class Sort {
    public static void selectionSort(char[] a) {
        for (int i = 0; i < a.length; i++) {
            int minIdx = i;
            for (int k = i + 1; k < a.length; k++) {
                if (a[k] < a[minIdx]) {
                    minIdx = k;
                }
            }
            char tmp = a[i];
            a[i] = a[minIdx];
            a[minIdx] = tmp;
            // There’s an S.o.p statement here!
            System.out.println(Arrays.toString(a));
        }
    }

    public static void main(String[] a) {
        char[] b = {'a', 'f', 'b', 'd', 'u'};
        System.out.println(Arrays.toString(b));
        selectionSort(b);
        System.out.println(Arrays.toString(b));
    }
}
```
4. (14 points) Method overloading and variable scopes
What is the output of following Java program?

class Scope {
    public static int z = 10;
    public static String a = "10";

    public static void main(String[] args) {
        System.out.println(z);
        {
            System.out.println(z);
        }
        {
            System.out.println(z);
            boolean z = true;
            System.out.println(z + " = " + Scope.z + "?");
        }
        System.out.println(z(a + z));

        z = 20;
        {
            double z = 10;
            z = z( (int) z);
            System.out.println(z);
        }

        z = z( z + Integer.parseInt(a) );
        System.out.println(z);

        z = 10;
        for (int z = 10; z < 1000; z += Scope.z) {
            if (z % 200 == 10) {
                System.out.println(z);
            }
        }
        System.out.println(z);
    }
}

class Scope {
    public static int z(int z) {
        return z * z;
    }

    public static String z(String z) {
        return z + z;
    }
}
5. (9 points) 2-D arrays

a) The following code is a Java program that compiles and run. Please provide the output of the program (i.e. what values are printed to the console/terminal).

```java
public class Arr2D {
    public static void main(String[] args) {
        double[][] arr = new double[3][2];

        for (int k = 0; k < arr.length; k++) {
            for (int i = 0; i < arr[0].length; i++) {
                arr[k][i] = k + i;
            }
        }

        System.out.println(Arrays.toString(arr[1]));

        for (int a = 0; a < arr[0].length; a++) {
            for (int b = 0; b < a; b++) {
                System.out.print(arr[b][a] + " ");
            }
            System.out.println();
        }

        System.out.println(Arrays.toString(arr[0]));
    }
}
```

Output:
6. (12 points) Methods (Write both header and method body)

a) (6 pts) Write a complete method named `getShortestStr` that accepts an array of Strings, and returns the String with the shortest length. If the array does not contain any String, return null.

```java
/* Write your method here */
```

b) (6 pts) Write a complete method named `checkFor2And3` that accepts an array of floating type numbers (double), and returns true if the array contains both numbers 2.0 and 3.0; otherwise, return false.

Example:

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Return</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkFor2And3({1, 8, 0, 2, 5})</td>
<td>false</td>
<td>Only 2 appears.</td>
</tr>
<tr>
<td>checkFor2And3({1, 0, 2, 3})</td>
<td>true</td>
<td>Both 2 and 3 appear.</td>
</tr>
<tr>
<td>checkFor2And3({0, 4, 6, 5, 7, 6})</td>
<td>false</td>
<td>Neither number 2 nor 3 appears.</td>
</tr>
</tbody>
</table>

/* Write your method here */
7. (12 points) Methods (Write both header and method body)

a) (9pts) Write a complete method named `numCommon` that accepts 2 arrays of integers, and returns the number of elements appearing in both arrays. Assume for each array, each element appears only once.

Hint: Use nested loops!

For example:

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Return</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>numCommons({1, 8, 5}, {2, 9, 8})</code></td>
<td>1</td>
<td>Only 8 appears in both arrays.</td>
</tr>
<tr>
<td><code>numCommons({0, 9, 5, 2}, {1, 2, 9, 0})</code></td>
<td>3</td>
<td>0, 2 and 9 appear in both arrays</td>
</tr>
<tr>
<td><code>numCommons({1, 2, 11}, {9})</code></td>
<td>0</td>
<td>No element appears in both arrays.</td>
</tr>
</tbody>
</table>

b) (3 pts) Write a complete method named `numNotCommon` that accepts 2 arrays of integers, and returns the number of elements appearing only one array both not the other; that is the total of elements appearing only in the first array but not the second, and elements appearing only in the second array but not the first.

You can (should) use a method call to `numCommon` from part a. You can assume that the method `numCommon` works as described.

e.g.

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Return</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>numCommons({1, 8, 5}, {2, 9, 8})</code></td>
<td>4</td>
<td>1, 2, 5, 9 are not common elements.</td>
</tr>
<tr>
<td><code>numCommons({0, 9, 5, 2}, {1, 2, 9, 0})</code></td>
<td>2</td>
<td>1 and 5 are not common elements</td>
</tr>
<tr>
<td><code>numCommons({1, 2, 11}, {9})</code></td>
<td>4</td>
<td>1, 2, 9, 11 are not common elements.</td>
</tr>
</tbody>
</table>
8. (10 points) 2D array

Write a complete method named `getTotals` that accepts a two-dimensional array of integers and also a boolean parameter `useRows`.

If `useRows` is true, the method should return a new array containing the sums of elements in each row.
If `useRows` is false, the method should return a new array containing the sums of elements in each column.

Example:

If \( a = \begin{pmatrix} 1, 2, 3, 4 \\ 2, 5, 1, 7 \end{pmatrix} \)

<table>
<thead>
<tr>
<th>Method Call on ( a )</th>
<th>Returned array</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getTotals(a, true)</code></td>
<td>{10, 15}</td>
</tr>
<tr>
<td><code>getTotals(a, false)</code></td>
<td>{3, 7, 4, 11}</td>
</tr>
</tbody>
</table>

/* Write your method here */
Bonus question (+3 pts):
What would be the output of the following program?

```java
public class Bonus {
    int a = 10;
    a = a++ + a;
    int b = 10;
    b = b + b++;
    System.out.println(a + " and " + ++b);
}
```

Output:

```
ASCII Table
```

Note that uppercase: 65 <= x <= 90
Note that lowercase 97 <= x <= 122
Difference between A (65) and a (97) is 32!
Difference between Q (81) and q (113) is 32!